Springer Tracts in Advanced Robotics

Volume 142

The Springer Tracts in Advanced Robotics (STAR) publish new developments and advances in the fields of robotics research, rapidly and informally but with a high quality. The intent is to cover all the technical contents, applications, and multidisciplinary aspects of robotics, embedded in the fields of Mechanical Engineering, Computer Science, Electrical Engineering, Mechatronics, Control, and Life Sciences, as well as the methodologies behind them. Within the scope of the series are monographs, lecture notes, selected contributions from specialized conferences and workshops, as well as selected PhD theses.

Special offer: For all clients with a print standing order we offer free access to the electronic volumes of the Series published in the current year.

Indexed by SCOPUS, DBLP, EI Compendex, zbMATH, SCImago.
All books published in the series are submitted for consideration in Web of Science.

More information about this series at http://www.springer.com/series/5208

Peter Corke

Robotic Vision

Fundamental Algorithms in MATLAB®

Springer

Peter Corke
School of Electrical Engineering
and Robotics
Queensland University of Technology
Brisbane, QLD, Australia

ISSN 1610-7438 ISSN 1610-742X (electronic)
Springer Tracts in Advanced Robotics
ISBN 978-3-030-79174-2 ISBN 978-3-030-79175-9 (eBook)
https://doi.org/10.1007/978-3-030-79175-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To my family Phillipa, Lucy and Madeline for their indulgence and support;
my parents Margaret and David for kindling my curiosity;
and to Lou Paul who planted the seed that became this book.

Foreword

At the dawn of the century's third decade, robotics is reaching an elevated level of maturity and continues to benefit from the advances and innovations in its enabling technologies. These all are contributing to an unprecedented effort to bringing robots to human environment in hospitals and homes, factories and schools; in the field for robots fighting fires, making goods and products, picking fruits and watering the farmland, saving time and lives. Robots today hold the promise for making a considerable impact in a wide range of real-world applications from industrial manufacturing to healthcare, transportation, and exploration of the deep space and sea. Tomorrow, robots will become pervasive and touch upon many aspects of modern life.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

This is a refined remake of the volume of the second edition of *Robotics, Vision and Control – Fundamental Algorithms in MATLAB®* by Peter Corke in 2017. The work now comes in two split volumes: one devoted to *Robotics and Control*, and the other to *Robotic Vision*. The first volume contains material from the first nine chapters of the previous single volume, covering: foundations on pose, time, and motion; mobile robots with navigation and localization; kinematics, dynamics, and control of robot manipulators. On the other hand, the second volume contains material from the first two chapters and the tenth to fourteenth chapters of the previous single volume, covering: foundations on pose, computer vision, image processing and feature extraction; image formation and multiple images for the geometry of vision.

The outcome is a two-volume handy set which is confirmed to be shining in our STAR series!

Naples, Italy and Stanford, USA
November 2020

Bruno Siciliano and *Oussama Khatib*
STAR Editors

Foreword to the Second Edition

Once upon a time, a very thick document of a dissertation from a faraway land came to me for evaluation. *Visual robot control* was the thesis theme and *Peter Corke* was its author. Here, I am reminded of an excerpt of my comments, which reads, *this is a masterful document, a quality of thesis one would like all of one's students to strive for, knowing very few could attain – very well considered and executed.*

The connection between robotics and vision has been, for over two decades, the central thread of Peter Corke's productive investigations and successful developments and implementations. This rare experience is bearing fruit in this second edition of his book on *Robotics, Vision, and Control*. In its melding of theory and application, this second edition has considerably benefited from the author's unique mix of academic and real-world application influences through his many years of work in robotic mining, flying, underwater, and field robotics.

There have been numerous textbooks in robotics and vision, but few have reached the level of integration, analysis, dissection, and practical illustrations evidenced in this book. The discussion is thorough, the narrative is remarkably informative and accessible, and the overall impression is of a significant contribution for researchers and future investigators in our field. Most every element that could be considered as relevant to the task seems to have been analyzed and incorporated, and the effective use of Toolbox software echoes this thoroughness.

The reader is taken on a realistic walkthrough the fundamentals of mobile robots, navigation, localization, manipulator-arm kinematics, dynamics, and joint-level control, as well as camera modeling, image processing, feature extraction, and multi-view geometry. These areas are finally brought together through extensive discussion of visual servo system. In the process, the author provides insights into how complex problems can be decomposed and solved using powerful numerical tools and effective software.

The *Springer Tracts in Advanced Robotics (STAR)* is devoted to bringing to the research community the latest advances in the robotics field on the basis of their significance and quality. Through a wide and timely dissemination of critical research developments in robotics, our objective with this series is to promote more exchanges and collaborations among the researchers in the community and contribute to further advancements in this rapidly growing field.

Peter Corke brings a great addition to our STAR series with an authoritative book, reaching across fields, thoughtfully conceived and brilliantly accomplished.

Oussama Khatib
Stanford, California
October 2016

Preface

Tell me and I will forget.
Show me and I will remember.
Involve me and I will understand.
Chinese proverb

Simple things should be simple,
complex things should be possible.
Alan Kay

Digital images are everywhere – most of us have collections of thousands of digital images captured on a variety of devices. Each image is rich in information about the scene it recorded – implicit in every pixel is information about the 3D structure of the scene, the materials which comprise it and the lighting conditions at the time. In our everyday life we use our sense of vision to help us perform a huge variety of tasks: recognizing people, reading text, manipulating objects and navigating. Some simple visual skills can now be performed by machines, for example, cameras can tell if we're smiling, social media platforms and customs barriers can recognize individual faces; and we can search our online photo albums using text keywords. Robots need mastery of visual skills if they are to meet their potential as helpful and ubiquitous machines in our everyday lives.

Robots are data-driven machines – they acquire data, process it and take action based on it. The data comes from many sensors including cameras. An image comprises millions of pixels and the amount of data that needs to be processed, in real-time, is massive.

Progress robotic vision has been, and continues to be, driven by more effective ways to process that data. This has been achieved through new and more efficient algorithms, as well as the dramatic increase in computational power that follows Moore's law.◀ When I started in robotics and vision in the mid 1980s, see Fig. 0.1, the IBM PC had been recently released – it had a 4.77 MHz 16-bit microprocessor and 16 kbytes (expandable to 256 k) of memory. Over the intervening 30+ years computing power has perhaps doubled 20 times which is an increase by a factor of over one million. In the last 5 years, graphical processing units, or GPUs, have come to the fore and powered a new wave of data-driven algorithms most notably deep learning◀.

"Computers in the future may weigh no more than 1.5 tons." Popular Mechanics, forecasting the relentless march of science, 1949

The topic of deep learning is outside the scope of this book.

Over the fairly recent history of robotic vision a very large body of algorithms has been developed to efficiently solve large-scale problems in scene understanding◀, motion estimation, and 3D reconstruction – a significant, tangible, and collective achievement of the research community. However its sheer size and complexity presents a very real barrier to somebody new entering the field. Given so many algorithms from which to choose, a real and important question is:

"Understanding" is a word with strong connotations and it is a philosophical question as to whether a machine can ever "understand". In this context, we use a weak interpretation of "understanding" which encompasses recognizing objects and perhaps activities.

What is the right algorithm for my particular problem?

One strategy would be to try a few different algorithms and see which works best for the problem at hand, but this is not trivial and leads to the next question:

How can I evaluate algorithm X on my own data without spending days coding and debugging it from the original research papers?

Two developments come to our aid. The first is the availability of general purpose mathematical software which makes it easy to prototype algorithms. There are commercial packages such as MATLAB®, Mathematica®, Maple® and MathCad®◀, as well as open source projects such as SciLab, Octave, and Matplotlib. All these tools deal naturally and effortlessly with vectors and matrices, can create complex and beautiful

Respectively the trademarks of The MathWorks Inc., Wolfram Research, MapleSoft and PTC.

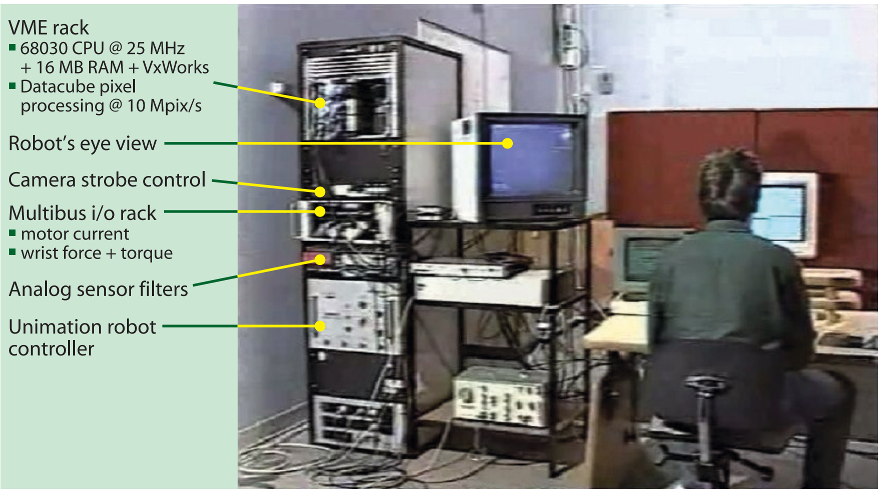

Fig. 0.1.
Once upon a time a lot of equipment was needed to do vision-based robot control. The author with a large rack full of real-time image processing and robot control equipment (1992)

graphics, and can be used interactively or as a programming environment. The second is the open-source movement. Many algorithms developed by researchers are available in open-source form. They might be coded in one of the general purpose mathematical languages just mentioned, or written in a mainstream language like C, C++, Java or Python.

For more than twenty years I have been part of the open-source community and maintained two open-source MATLAB Toolboxes: one for machine vision and one for robotics. They date back to my own Ph.D. work and have evolved since then, growing features and tracking changes to the MATLAB language. They form the basis of this book.

These Toolboxes have some important virtues. Firstly, they have been around for a long time and have been used by many people for many different problems so the code can be accorded some level of trust. New algorithms, or even the same algorithms coded in new languages or executing in new environments, can be compared against implementations in the Toolbox.

» allow the user to work with real problems, not just trivial examples

Secondly, they allow the user to work with real problems, not just trivial examples. For real images with millions of pixels the computation required is beyond unaided human ability. Thirdly, they allow us to gain insight which can otherwise get lost in the complexity. We can rapidly and easily experiment, play *what if* games, and depict the results graphically using the powerful 2D and 3D graphical display tools of MATLAB. Fourthly, the Toolbox code makes many common algorithms tangible and accessible. You can read the code, you can apply it to your own problems, and you can extend it or rewrite it. It gives you a "head start" as you begin your journey into robotic vision.

The book takes a conversational approach, weaving text, mathematics and code examples into a narrative. I wanted to show how complex problems can be decomposed and solved using just a few simple lines of code. More formally this is an inductive learning approach, going from specific and concrete examples to the more general.

» show how complex problems can be decomposed and solved

The topics covered in this book are based on my own interests but also guided by real problems that I observed over many years as a practitioner of both robotics and computer vision. I want to give you a flavor of what robotic vision is about and what it can do – consider it a grand tasting menu. I hope that by the end of this book you will share my enthusiasm for these topics.

» consider it a grand tasting menu

I was particularly motivated to present a solid introduction to computer vision for roboticists. The treatment of vision in robotics textbooks tends to concentrate on simple binary vision techniques. In this book we will cover a broad range of topics including color vision, advanced segmentation techniques, image warping, stereo vision, motion estimation, bundle adjustment, visual odometry and image retrieval. We also cover nonperspective imaging using fisheye lenses, catadioptric optics and the emerging area of light-field cameras. These topics are growing in importance for robotics but are not commonly covered. Vision is a powerful sensor, and roboticists should have a solid grounding in modern fundamentals.

» software is a first-class citizen in this book

This book is unlike other text books, and deliberately so. Firstly, software is a first-class citizen in this book. Software is a tangible instantiation of the algorithms described – it can be read and it can be pulled apart, modified and put back together again. There are a number of classic books that use software in an illustrative fashion and have influenced my approach, for example *LaTeX: A document preparation system* (Lamport 1994), *Numerical Recipes in C* (Press et al. 2007), *The Little Lisper* (Friedman et al. 1987) and *Structure and Interpretation of Classical Mechanics* (Sussman et al. 2001). Over 700 examples in this book illustrate how the Toolbox software can be used and generally provide *instant gratification* in just a couple of lines of MATLAB code.

» instant gratification in just a couple of lines of MATLAB code

Secondly, building the book around MATLAB and the Toolbox means that we are able to tackle more realistic and more complex problems than other books.

» this book provides a complementary approach

The emphasis on software and examples does not mean that rigor and theory are unimportant – they are very important, but this book provides a complementary approach. It is best read in conjunction with other texts which do offer rigor and theoretical nourishment. The end of each chapter has a section on further reading and provides pointers to relevant textbooks, key papers and online resources. I try hard to use the least amount of mathematical notation required, but if you seek deep mathematical rigor this may not be the book for you.

The Toolbox also includes some great open-source software and I am grateful to the following for code that has been incorporated into the Machine Vision Toolbox: RANSAC by Peter Kovesi; pose estimation by Francesco Moreno-Noguer, Vincent Lepetit, and Pascal Fua; color space conversions by Pascal Getreuer; numerical routines for geometric vision by various members of the Visual Geometry Group at Oxford (from the web site of the book by Hartley and Zisserman 2003); *k*-means, SIFT and MSER algorithms from the wonderful VLFeat suite (vlfeat.org) by Andrea Vedaldi and Brian Fulkerson; graph-based image segmentation software by Pedro Felzenszwalb; and the OpenSURF feature detector by Dirk-Jan Kroon, and the Camera Calibration Toolbox by Jean-Yves Bouguet.

As I wrote I became fascinated by the mathematicians, scientists and engineers whose work, hundreds of years ago, underpins the science of robotic vision today. Some of their names have become adjectives like Gaussian, Laplacian or Cartesian or nouns like Jacobian. They are interesting characters from a distant era when science was a hobby and their day jobs were as doctors, alchemists, gamblers, astrologers, philosophers or mercenaries. In order to know whose shoulders we are standing on I have included small vignettes about the lives of some of these people – a smattering of history as a backstory.

In my own career I have had the good fortune to work with many wonderful people who have inspired and guided me. Long ago at the University of Melbourne John Anderson fired my interest in control and Graham Holmes tried with mixed success to have me "think before I code". Early on I spent a life-direction-changing ten months working with Richard (Lou) Paul in the GRASP laboratory at the University of Pennsylvania in the period 1988–1989. The genesis of the Toolboxes was my Ph.D. research (1991–1994) and my advisors Malcolm Good (University of Melbourne) and Paul Dunn (CSIRO) asked me good questions and guided my research. Laszlo Nemes (CSIRO) provided great wisdom about life and the ways of organizations, and encouraged me to publish and to open-source my software. Much of my career was spent at CSIRO where I had the privilege and opportunity to work on a diverse range of real robotics and vision projects and to work with a truly talented set of colleagues and friends. Part way through writing the first edition I joined the Queensland University of Technology which made time available to complete that work, and in 2015 sabbatical leave to complete the second.

Many people have helped me in my endeavor and I thank them. I was generously hosted for periods of productive writing at Oxford (both editions) by Paul Newman, and at MIT (first edition) by Daniela Rus. Daniela, Paul and Cédric Pradalier made constructive suggestions and comments on early drafts of that edition. For the second edition I was helped by comments on draft chapters by: Tim Barfoot, Dmitry Bratanov, Duncan Campbell, Donald Dansereau, Tom Drummond, Malcolm Good, Peter Kujala, Obadiah Lam, Jörn Malzahn, Felipe Nascimento Martins, Ajay Pandey, Cédric Pradalier, Dan Richards, Daniela Rus, Sareh Shirazi, Surya Singh, Ryan Smith, Ben Talbot, Dorian Tsai and Ben Upcroft; and assisted with wisdom and content by: François Chaumette, Donald Dansereau, Kevin Lynch, Robert Mahony and Frank Park.

I have tried my hardest to eliminate errors but inevitably some will remain. Please email bug reports to me at rvc@petercorke.com as well as suggestions for improvements and extensions.

Writing the second edition was financially supported by EPSRC Platform Grant EP/M019918/1, QUT Science & Engineering Faculty sabbatical grant, QUT Vice Chancellor's Excellence Award, QUT Robotics and Autonomous Systems discipline and the ARC Centre of Excellence for Robotic Vision (grant CE140100016).

Over both editions I have enjoyed wonderful support from MathWorks, through their author program, and from Springer. My Springer editor Thomas Ditzinger has been a great supporter of this project and Armin Stasch, with enormous patience and dedication in layout and typesetting, has transformed my untidy ideas into a thing of beauty.

Finally, my deepest thanks are to Phillipa who has supported me and "the book" with grace and patience for a very long time and in many different places – without her this book could never have been written.

Peter Corke
Brisbane,
Queensland
August 2018

Note on the Second Edition

The revision principle was to keep the good (narrative style, code as a first-class citizen, soft plastic cover) and eliminate the bad (errors and missing topics). There were more errors than I would have liked and I thank everybody who submitted errata and suggested improvements.

The first edition was written before I taught in the university classroom or created the MOOCs, which is the inverse of the way books are normally developed. Preparing for teaching gave me insights into better ways to present some topics, particularly around pose representation, robot kinematics and dynamics so the presentation has been adjusted accordingly.

New content includes matrix exponential notation; the basics of screw theory and Lie algebra; inertial navigation; differential steer and omnidirectional mobile robots; a deeper treatment of SLAM systems including scan matching and pose graphs; greater use of MATLAB computer algebra; operational space control; deeper treatment of manipulator dynamics and control; visual SLAM and visual odometry; structured light; bundle adjustment; and light-field cameras.

In the first edition I shied away from Lie algebra, matrix exponentials and twists but I think it's important to cover them. The topic is deeply mathematical and I've tried to steer a middle ground between hardcore algebraic topology and the homogenous transformation only approach of most other texts, while also staying true to the overall approach of this book.

All MATLAB generated figures have been regenerated to reflect recent improvements to MATLAB graphics and all code examples have been updated as required and tested, and are available as MATLAB Live Scripts.

The second edition of the book is matched by new major releases of my Toolboxes: Robotics Toolbox (release 10) and the Machine Vision Toolbox (release 4). These newer versions of the toolboxes have some minor incompatibilities with previous releases of the toolboxes, and therefore also with the code examples in the first edition of the book.

Note on this Edition

This book is essentially Parts I and IV of the second edition of Robotics, Vision & Control. The previous Part IV has been rearranged and split into two parts, and the content related to robotics and vision-based control has been omitted.

Contents

Nomenclature

The notation used in robotic vision varies considerably across books and research papers. The symbols used in this book, and their units where appropriate, are listed below. Some symbols have multiple meanings and their context must be used to disambiguate them.

Notation	Description
x^*	desired value of x
x^+	predicted value of x
$x^\#$	measured, or observed, value of x
$\hat{x}$	estimated value of x
$\bar{x}$	mean of x or relative value
$x\langle k \rangle$	k^{th} element of a time series
$\boldsymbol{v}$	a vector
$\hat{\boldsymbol{v}}$	a unit-vector parallel to $\boldsymbol{v}$
$\tilde{\boldsymbol{v}}$	homogeneous representation of vector $\boldsymbol{v}$
$v[i]$	i^{th} element of vector $\boldsymbol{v}$
v_x	a component of a vector
$\boldsymbol{A}$	a matrix
$\boldsymbol{A}[i,j]$	the element (i,j) of $\boldsymbol{A}$
$\boldsymbol{A}_{i,j}$	the element (i,j) of $\boldsymbol{A}$
$f(x)$	a function of x
$F_x(x)$	the derivative $\partial f / \partial x$
$F_{xy}(x,y)$	the derivative $\partial^2 f / \partial x \partial y$
$\mathring{q}$	unit quaternion, $\mathring{q} \in \mathbb{S}^3$
$\boldsymbol{0}_{m \times n}$	an $m \times n$ matrix of zeros
$\boldsymbol{1}_{m \times n}$	an $m \times n$ matrix of ones

Symbol	Description	Unit
$\boldsymbol{C}$	camera matrix, $\boldsymbol{C} \in \mathbb{R}^{3\times4}$	
E	illuminance (lux)	lx
f	focal length	m
$\boldsymbol{f}$	vector of image features	
$\mathbb{H}$	the set of all quaternions (H for Hamilton)	
$\boldsymbol{I}_{n\times n}$	$n \times n$ identity matrix	
$\boldsymbol{J}$	Jacobian matrix	
${}^{A}\boldsymbol{J}_{B}$	Jacobian transforming velocities in frame B to frame A	
k, K	constant	
$\boldsymbol{K}$	camera calibration matrix	
L	luminance (nit)	nt
$N(\mu, \sigma^2)$	a normal (Gaussian) distribution with mean μ and standard deviation σ	
$\mathbf{p}$	an image plane point, $\mathbf{p} \in \mathbb{R}^2$	
$\mathbf{P}$	a world point, $\mathbf{P} \in \mathbb{R}^3$	
$\mathbb{P}^2$	projective space of all 2-D points, a 3-tuple	
$\mathbb{P}^3$	projective space of all 3-D points, a 4-tuple	
$\boldsymbol{q}$	generalized coordinates, configuration $\boldsymbol{q} \in \mathcal{C}$	m, rad
$\boldsymbol{R}$	an orthonormal rotation matrix, $\boldsymbol{R} \in \mathbf{SO}(2)$ or $\mathbf{SO}(3)$	
$\mathbb{R}$	set of real numbers	
$\mathbb{R}^2$	set of all 2-D points	
$\mathbb{R}^3$	set of all 3-D points	
$\mathbb{S}^1$	unit circle, set of angles $[0, 2\pi)$	
$\mathbb{S}^n$	unit sphere embedded in $\mathbb{R}^{n+1}$	
$\mathbf{se}(n)$	Lie algebra for $\mathbf{SE}(n)$, an $\mathbb{R}^{(n+1)\times(n+1)}$ augmented skew-symmetric matrix	
$\mathbf{so}(n)$	Lie algebra for $\mathbf{SO}(n)$, an $\mathbb{R}^{n\times n}$ skew-symmetric matrix	
$\mathbf{SE}(n)$	special Euclidean group, the set of all poses in n dimensions, represented by an $\mathbb{R}^{(n+1)\times(n+1)}$ homogeneous transformation matrix	
$\mathbf{SO}(n)$	special orthogonal group, the set of all orientations in n dimensions, represented by an $\mathbb{R}^{n\times n}$ orthogonal matrix	
S	twist in 3 dimensions, $S \in \mathbb{R}^6$	
t	time	s
T	temperature	K
T	optical transmission	m^{-1}
$\boldsymbol{T}$	homogeneous transformation, $T \in \mathbf{SE}(2)$ or $\mathbf{SE}(3)$	
${}^{A}\boldsymbol{T}_{B}$	homogeneous transform representing frame $\{B\}$ with respect to frame $\{A\}$. If A is not given then assumed relative to world coordinate frame 0. Note that ${}^{A}\boldsymbol{T}_{B} = ({}^{B}\boldsymbol{T}_{A})^{-1}$	
u, v	camera image plane coordinates	pixels
u_0, v_0	coordinates of the principal point	pixels
$\bar{u}, \bar{v}$	normalized image plane coordinates, relative to the principal point	m
v	velocity	$\mathrm{m\ s}^{-1}$
$\boldsymbol{v}$	velocity vector	$\mathrm{m\ s}^{-1}$
X, Y, Z	Cartesian coordinates	
$\bar{x}, \bar{y}$	normalized image-plane coordinates	
$\mathbb{Z}$	set of all integers	
$\mathbb{Z}^+$	the set of all integers greater than zero	

Symbol	Description	Unit
ϕ	luminous flux (lumens)	lm
$\boldsymbol{\Gamma}$	3-angle representation of rotation, $\boldsymbol{\Gamma} \in \mathbb{R}^3$	rad
θ	angle	rad
$\theta_r, \theta_p, \theta_y$	roll pitch yaw angles	rad
λ	wavelength	m
λ	an eigenvalue	
ν	spatial velocity, $\nu = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z) \in \mathbb{R}^6$	m s^{-1}, rad s^{-1}
ξ	abstract representation of Cartesian pose (pronounced ksi)	
${}^A\xi_B$	abstract representation of relative pose, frame $\{B\}$ with respect to frame $\{A\}$ or rigid-body motion from frame $\{A\}$ to $\{B\}$	
π	mathematic constant	
$\boldsymbol{\pi}$	a plane	
ρ_w, ρ_h	pixel width and height	m
σ	standard deviation	
$\boldsymbol{\Sigma}$	Lie algebra $\boldsymbol{\Sigma} = [\cdot] \in \mathbf{se}(3)$	
ω	rotational rate	rad s^{-1}
$\boldsymbol{\omega}$	angular velocity vector	rad s^{-1}
$\boldsymbol{\Omega}$	Lie algebra $\boldsymbol{\Omega} = [\cdot]_\times \in \mathbf{so}(3)$	

Operator	Description	MATLAB
$\lVert\cdot\rVert$	norm, or length, of vector: $\mathbb{R}^n \mapsto \mathbb{R}$	`norm, .norm`
$\boldsymbol{v}_1 \cdot \boldsymbol{v}_2$	dot, or inner, product, also $\boldsymbol{v}_1^T\boldsymbol{v}_2$: $\mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$	`dot`
$\boldsymbol{v}_1 \times \boldsymbol{v}_2$	cross, or vector, product: $\mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$	`cross`
$\boldsymbol{A}^{-1}$	inverse of $\boldsymbol{A}$: $\mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$	`inv`
$\boldsymbol{A}^{+}$	pseudo-inverse of $\boldsymbol{A}$: $\mathbb{R}^{n\times m} \mapsto \mathbb{R}^{m\times n}$	`pinv`
$\boldsymbol{A}^{*}$	adjugate of $\boldsymbol{A} \mapsto \det(\boldsymbol{A})\boldsymbol{A}^{-1}$, $\mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$	
$\boldsymbol{A}^{T}$	transpose of $\boldsymbol{A}$: $\mathbb{R}^{n\times m} \mapsto \mathbb{R}^{m\times n}$	`'`
$\boldsymbol{A}^{-T}$	transpose of inverse $\boldsymbol{A} \mapsto (\boldsymbol{A}^T)^{-1} = (\boldsymbol{A}^{-1})^T$, $\mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$	
$\bullet$	transform a point (coordinate vector) by a relative pose: $\mathbf{SE}(n) \times \mathbb{R}^n \mapsto \mathbb{R}^n$	`*`
$\oplus$	composition: $\mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n) \times \mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n) \mapsto \mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n)$	`*`
$\ominus$	composition with inverse: $\mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n) \times \mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n) \mapsto \mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n)$	`/`
$\ominus$	unary inverse: $\mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n) \mapsto \mathbf{S}^{\mathbf{O}}_{\mathbf{E}}(n)$	`.inv`
$\Delta(\cdot)$	maps incremental pose change to differential motion: $\mathbf{SE}(3) \mapsto \mathbb{R}^6$	`tr2delta`
$\Delta^{-1}(\cdot)$	maps differential motion to incremental pose change: $\mathbb{R}^6 \mapsto \mathbf{SE}(3)$	`delta2tr`
$\mathscr{R}_i(\theta)$	pure rotation about axis i: $\mathbb{R} \mapsto \mathbf{SE}(3)$	`SE3.rotx|y|z`
$\mathscr{R}(\boldsymbol{\omega})$	pure rotation by $\lVert\boldsymbol{\omega}\rVert$ about $\boldsymbol{\omega}$: $\mathbb{R}^3 \mapsto \mathbf{SE}(3)$	`SE3.angvec`
$\mathscr{T}_i(d)$	pure translation along axis i: $\mathbb{R} \mapsto \mathbf{SE}(2), \mathbf{SE}(3)$	`SE2, SE3`
$\mathscr{T}(\boldsymbol{t})$	pure translation by vector: $\mathbb{R}^n \mapsto \mathbf{SE}(n)$	`SE2, SE3`
$[\cdot]_t$	translational component of pose: $\mathbf{SE}(n) \mapsto \mathbb{R}^n$	`.t`
$[\cdot]_R$	rotational component of pose: $\mathbf{SE}(n) \mapsto \mathbb{R}^{n\times n}$	`.R`
$[\cdot]_\times$	skew-symmetric matrix: $\mathbb{R} \mapsto \mathbf{so}(2)$, $\mathbb{R}^3 \mapsto \mathbf{so}(3)$	`skew`
$\vee_\times(\cdot)$	*unpack* skew-symmetric matrix: $\mathbf{so}(2) \mapsto \mathbb{R}$, $\mathbf{so}(3) \mapsto \mathbb{R}^3$	`vex`
$[\cdot]$	augmented skew-symmetric matrix: $\mathbb{R}^3 \mapsto \mathbf{se}(2)$, $\mathbb{R}^6 \mapsto \mathbf{se}(3)$	`skewa`
$\vee(\cdot)$	*unpack* augmented skew-symmetric matrix: $\mathbf{se}(2) \mapsto \mathbb{R}^3$, $\mathbf{so}(3) \mapsto \mathbb{R}^6$	`vexa`
$\mathrm{Ad}(\cdot)$	adjoint representation: $\mathbf{SE}(3) \mapsto \mathbb{R}^{6\times 6}$	`.Ad`
$\mathrm{ad}(\cdot)$	logarithm of adjoint representation: $\mathbf{SE}(3) \mapsto \mathbb{R}^{6\times 6}$	`.ad`
$\circ$	quaternion (Hamiltonian) multiplication: $\mathbb{H} \times \mathbb{H} \mapsto \mathbb{H}$	`*`
$\mathring{v}$	pure quaternion: $\mathbb{R}^3 \mapsto \mathbb{H}$	`Quaternion.pure`
$\sim$	equivalence of representations	
$\simeq$	homogeneous coordinate equivalence	
$\ominus$	smallest angular difference between two angles on a circle: $\mathbb{S}^1 \times \mathbb{S}^1 \mapsto \mathbb{R}$	`angdiff`
$\mathcal{P}(\cdot)$	camera projection function: $\mathbb{R}^3 \mapsto \mathbb{R}^2$	`.project`
$\star$	convolution	`iconv`
$\otimes$	correlation	
$\equiv$	colormetric equivalence	
$\oplus$	morphological dilation	
$\ominus$	morphological erosion	
$\circ$	morphological opening	
$\bullet$	morphological closing	
$\{F\}$	coordinate frame F	
$[a, b]$	interval a to b inclusive	
(a, b)	interval a to b exclusive, not including a or b	
$[a, b)$	interval a to b, not including b	
$(a, b]$	interval a to b, not including a	

MATLAB® Toolbox Conventions

- A Cartesian coordinate, a point, is expressed as a column vector.
- A set of points is expressed as a matrix with columns representing the coordinates of individual points.
- A rectangular region is defined by its two opposite corners as a 2 × 2 matrix $[x_{\min}\ x_{\max};\ y_{\min}\ y_{\max}]$.
- Time series data is expressed as a matrix with rows representing time steps.
- A MATLAB matrix has subscripts (i, j) which represent row and column respectively. Image coordinates are written (u, v) so an image represented by a matrix $\boldsymbol{I}$ is indexed as $\boldsymbol{I}(v, u)$.
- Matrices with three or more dimensions are frequently used:
  - A color image has 3 dimensions: row, column, color plane.
  - A greyscale image sequence has 3 dimensions: row, column, index.
  - A color image sequence has 4 dimensions: row, column, color plane, index.

Common Abbreviations

2D	2-dimensional
3D	3-dimensional
n-tuple	A group of *n* numbers, it can represent a point of a vector